

IN THE CLAIMS

Please cancel claims 19-30 and 32 as follows:

1. (PREVIOUSLY PRESENTED) A method of transmitting a data segment in a data stream using a write module which implements a selected one of a plurality of versions of a streaming protocol wherein each subsequent version of the streaming protocol is additive to a previous version, the method comprising the steps of:
  - (a) outputting a first stream of data according to a first version of the streaming protocol;
  - (b) sequentially appending additional streams of data to the first stream of data according to each subsequent version of the streaming protocol up to and including the selected version, if the selected version of the streaming protocol is not the first version of the streaming protocol; and
  - (c) delimiting the data segment in the data stream using begin and end tags.
2. (ORIGINAL) The method of claim 1, further comprising the step of receiving the data segment from a data stream using a read module of the type which implements a second selected one of the plurality of versions of the streaming protocol, the receiving step including the steps of:
  - receiving the first stream of data;
  - if the second selected version is earlier than the first selected version, receiving each additional stream of data according to each subsequent version of the streaming protocol up to and including the second selected version, and disregarding any remaining data in the data segment;
  - if the second selected version is equal to or later than the first selected version, sequentially receiving the additional streams of data according to each subsequent version of the streaming protocol up to and including the second selected version; and
  - testing, prior to receiving each additional stream of data, whether an end of the data segment has been detected, and if so, terminating reception of the data segment prior to receiving the additional stream of data according to the second selected version.
3. (ORIGINAL) The method of claim 2, wherein the data segment is an object.

4. (ORIGINAL) The method of claim 3, wherein the data segment includes all of the data necessary to reconstruct the object; whereby the data stream is serial.

5. (ORIGINAL) The method of claim 3, wherein the testing step includes the step of initializing object data that is not received from the data stream to a default value.

6. (ORIGINAL) The method of claim 3, further comprising the steps of:  
transmitting an object type for the data segment; and  
receiving the object type, including allocating and initializing an object when receiving the data segment based upon the object type.

7. (ORIGINAL) The method of claim 2, wherein the read and write modules are resident on the same computer.

8. (ORIGINAL) The method of claim 2, wherein the read and write modules are resident on separate computers.

9. (CANCELED)

10. (PREVIOUSLY PRESENTED) The method of claim 1, wherein no additional tags are embedded in the data segment between the begin and end tags.

11. (ORIGINAL) The method of claim 1, further comprising the steps of:  
determining whether the data segment is stored in a current context for the data stream;  
if so, transmitting an alias tag in lieu of the data segment; and  
if not, storing the data segment in the current context.

12. (ORIGINAL) The method of claim 1, wherein the data stream is a non-random access data stream.

13. (PREVIOUSLY PRESENTED) A method of receiving a data segment from a data stream using a read module which implements a selected one of a plurality of versions of a streaming protocol wherein each subsequent version of the streaming protocol is additive to a previous version, the method comprising the steps of:

- (a) receiving a first stream of data according to a first version of the streaming protocol;
- (b) if the selected version of the streaming protocol is not the first version of the streaming protocol, sequentially receiving additional streams of data according to each subsequent version of the streaming protocol up to and including the selected version; and
- (c) testing, prior to receiving each additional stream of data, whether an end of the data segment has been detected, and if so, terminating reception of the data segment prior to receiving the additional stream of data according to the selected version.

14. (ORIGINAL) The method of claim 13, further comprising the step of, if the end of the data segment has not been detected upon receiving the additional stream of data according to the selected version, disregarding any remaining data in the data segment.

15. (ORIGINAL) The method of claim 14, further comprising the step of storing the data segment in a current context, including any disregarded data therefrom.

16. (ORIGINAL) The method of claim 13, wherein the data segment is an object.

17. (ORIGINAL) The method of claim 16, wherein the testing step includes the step of initializing object data that is not received from the data stream to a default value.

18. (ORIGINAL) The method of claim 16, further comprising the steps of:  
receiving an object type for the data segment; and  
allocating and initializing an object based upon the object type to build the object from the  
streams of data in the data segment.

19. - 30. (CANCELED)

31. (PREVIOUSLY PRESENTED) The method of claim 13, wherein the step of  
testing whether an end of the data segment has been detected comprises the step of testing for a  
premature end tag and terminating the reception of the data segment when a premature end tag is  
received.

32. (CANCELED)